EC3-1481CLDNA

3.5” 单板电脑带 CPU 和 LCD/CRT/USB/LAN/CF/Audio 接口

版本：A0

非常感谢您购买"EVOC"产品

在打开包装箱后请首先依据物件清单检查配件，若发现物件有所损坏或是有任何配件短缺的情况，请尽快与您的经销商联络。

þ **1** 块 **EC3-1481CLDNA** 工业级主板

þ **1** 本用户手册

þ **1** 条键盘／鼠标转接电缆

þ **1** 条 **IDE** 转接电缆

þ **1** 条 **AUDIO** 转接电缆

þ **5** 条 **COM** 转接电缆

þ **1** 张 **EVOC** 光碟（内含驱动程序和用户手册）

þ 备用跳线帽

声明

安全使用小常识

1. 产品使用前，务必仔细阅读产品说明书；
2. 对未准备安装的板卡，应将其保存在防静电保护袋中；
3. 在从防静电保护袋中拿出板卡前，应将手先置于接地金属物体上一会儿（比如 **10** 秒钟），以释放身体及手中的静电；
4. 在拿板卡时，需佩戴静电保护手套，并且应该养成只触及其边缘部分的习惯；
5. 为避免人体被电击或产品被损坏，在每次对主板、板卡进行拔插或重新配置时，须先关闭交流电源或将交流电源线从电源插座中拔掉；
6. 在需对板卡或整机进行搬动前，务必先将交流电源线从电源插座中拔掉；
7. 对整机产品，需增加 / 减少板卡时，务必先拔掉交流电源；
8. 当您需连接或拔除任何设备前，须确定所有的电源线事先已被拔掉；
9. 为避免频繁开关机对产品造成不必要的损伤，关机后，应至少等待 **30** 秒后再开机。

目　录

第一章

产品介绍

简介

EC3-1481CLDNA 全功能嵌入式单板电脑采用 **133MHz** 低功耗 **CPU**，集成了 **32M SDRAM** 内存、**CRT/LCD** 双显示控制芯片、**10/100Mbps** 网络接口、**CompactFlash** 电子盘接口、**IDE** 接口、六串口、两个 **USB** 接口、看门狗定时器、**PCI** 声卡、**PS/2** 键盘鼠标接口、数字 **I/O** 接口、**PC104** 扩展总线。

EC3-1481CLDNA 采用低功耗 **CPU**，在 **-40**℃ **~85**℃范围内无须风扇可正常工作。板载 **TP6508** 显示芯片，配以 **1MB** 显示内存，支持 **CRT+LCD** 双显示，**LCD** 接口可支持 **TFT**、**DSTN** 等多种 **LCD** 屏。支持 **Dos,WinCE,Linux** 等操作系统。

EC3-1481CLDNA 以其全集成性能、丰富的接口功能以及高可靠性能，可广泛应用于自动化控制、仪器仪表、智能产品等各种嵌入式领域。

订购信息

型号	描述
EC3-1481CLDNA	3.5” 单板电脑带 CPU 和 LCD /CRT/USB/LAN/CF/Audio 接口

环境与机械尺寸

- 工作环境

  温度：**-40℃～85℃**　　　湿度：**5%～95%**（非凝结状态）

- 贮存环境

  温度：**-40℃～85℃**　　　湿度：**5%～95%**（非凝结状态）

- 尺寸：**146㎜×102㎜**

微处理器（CPU）

在板集成单芯片低功耗处理器**,**最高频率 **133MHZ**。

系统存储器（System Memory）

在板集成 **32MB SDRAM**。

IDE 功能

ITE8211F 控制器**,**一个 **44-pin IDE** 接口。

CF 卡

提供一个 **CompactFlash** 插座。

USB 功能

2 个 **USB** 接口，符合 **USB Ver2.0** 标准。

显示功能

TOPRO TP6508 图形控制芯片，**1MB** 显存，同时支持 **VGA** 和 **LCD** 显示**,** 可支持 **TFT/DSTN** 等类型的屏。

网络功能（LAN）

10M/100M 以太网控制器，**CPU** 内部集成 **MAC+**外部 **PHY**。

I/O 功能

两个 **USB2.0** 接口，六个 **RS-232**，其中一个为 **RS232/485** 可选，一个 **PS/2** 键盘**/**鼠标接口。

可编程 I/O

8 位输入输出＋**7** 位输出；或者定制成 **8** 位输入＋**8** 位输出；或者定制成 **8** 位输出（带回读）＋**8** 位输出。（**TTL** 电平）

Audio 支持

支持 **AC'97** 音频接口，**mic-in,line-in,speaker-out**。

Watchdog 功能

提供两个看门狗，一个是 **Super I/O** 的看门狗，一个是 **CPU** 内部的看门狗。

(1) Super I/O Watchdog

- 定时时间从 **1** 到 **255** 级可编程
- 可编程选择超时中断
- 可编程选择超时事件复位系统

(2) CPU Watchdog

- 定时时间 **8** 级可编程
- 可选择超时中断
- 可选择超时事件复位系统

扩充总线

一个 **PC/104** 总线。（不支持 **DMA**）

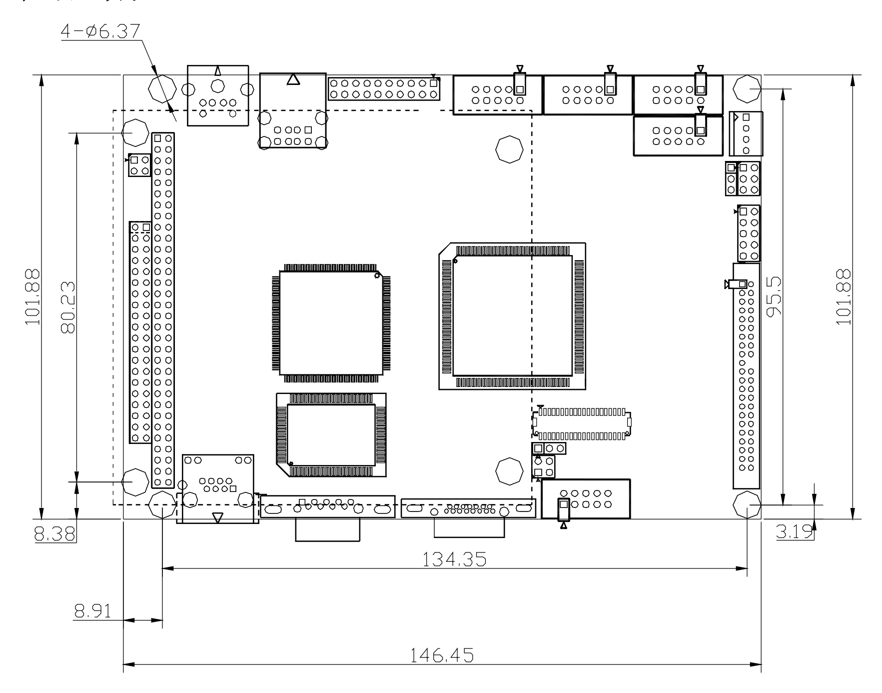
其他特性

单 5V 电源供电，软驱电源接口。

第二章

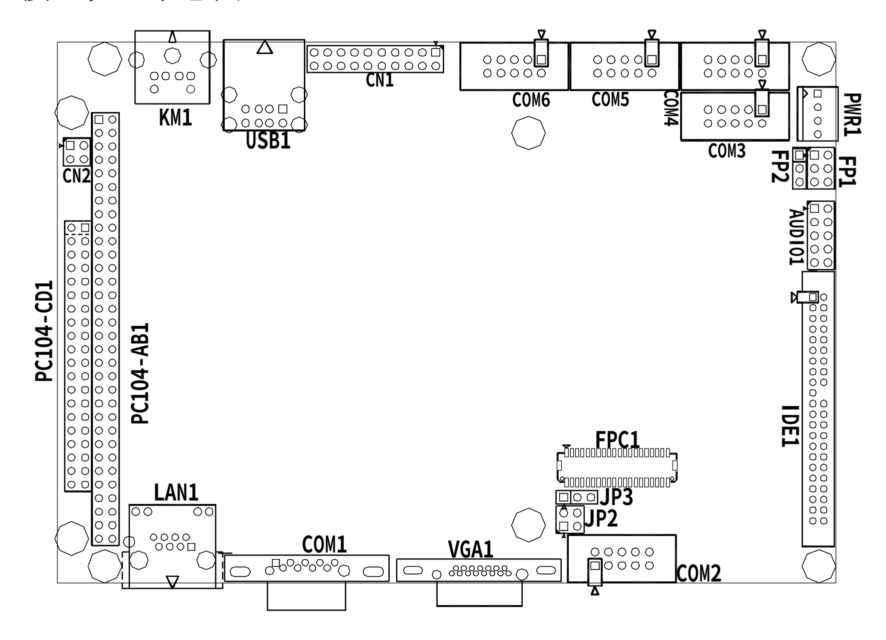
安装说明

产品外形



单位：**mm**

接口位置示意图

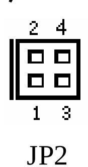

KM1
USB1
CN1
COM6
COM5
COM4
COM3
PWR1
CN2
FP2
FP1
AUDIO1
PC104-CD1
PC104-AB1
IDE1
FPC1
JP3
JP2
LAN1
COM1
VGA1
COM2

跳线功能设置

提示：如何识别跳线、接口的第一针脚

Ø　观察插头插座旁边的文字标记，会用“**1**”或加粗的线条或三角符号表示；看看背面的焊盘，方型焊盘为第一针脚；电缆上的红线或其它标记表示要与插座的第一脚相接。

(1) JP2:COM2 口模式选择



JP2

设置	功能
[1-2]短路，**[3-4]**断开	**RS232**
[1-2]断开，**[3-4]**短路	**RS485**

(2) JP3：LCD 屏工作电压选择

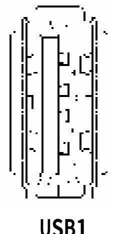
1 2 3

JP3

设置	电压
[1-2]短路	**3.3V (**默认设置**)**
[2-3]短路	**5V**

USB

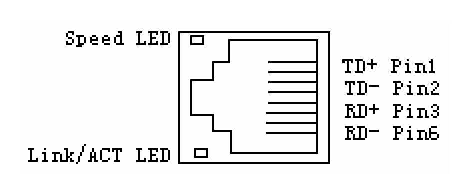
当与 **USB** 设备相连接时，只能以一个方向插入。下面是 **USB** 接口的各脚的定义：


USB1

管脚		信号名称
1	**5**	**VCC**
2	**6**	**USB Data-**
3	**7**	**USB Data +**
4	**8**	**GND**

网络接口

板上提供 **1** 个 **10/100Mbps** 以太网接口(**LAN1**)，以下给出了它的管脚安排和相应的输入插座。**Link/Act LED** 和 **Speed LED** 是以太网接口两边的绿色和黄色 **LED**。请参考以下每一个 **LED** 的状态描述：

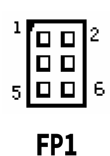


TD+,TD-：正/负发送数据信号。
RD+,RD-：正/负接收数据信号。
Speed LED：网络速度灯。
Link/Act LED：网络链路状态/活动状态灯。

LAN_LED0 是 **Link/Act LED**；**LAN_LED1** 是 **Speed LED**。请参考以下每一个 **LED** 的状态描述：

Link/Act LED（绿色）	指示状态	**Speed LED**（黄色）	指示状态
亮	已链接	亮	**100Mbps**
灭	未链接	灭	**10Mbps**
闪烁	正在收发数据		

状态指示接口



FP1

信号名称	管脚	管脚	信号名称
NC	1	2	NC
GND	3	4	RESET
IDE LED -	5	6	IDE LED +

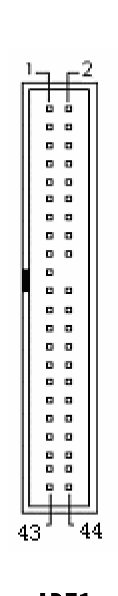
1
2
3

FP2

管脚	信号名称
1	Power LED +
2	NC
3	GND

IDE 接口

此接口为一 **44** 针，间距 **2** 毫米的硬盘接口。将一端连接到板上后，另一端接到硬盘。如果你安装两个硬盘，你就必须设置好主从跳线，请参考硬盘有关跳线主从方式的设置部分。(定义如下表)。



IDE1

管脚	信号名称	管脚	信号名称
1	复位	**2**	地
3	主机数据 7	**4**	主机数据 8
5	主机数据 6	**6**	主机数据 9
7	主机数据 5	**8**	主机数据 10
9	主机数据 4	**10**	主机数据 11
11	主机数据 3	**12**	主机数据 12
13	主机数据 2	**14**	主机数据 13
15	主机数据 1	**16**	主机数据 14
17	主机数据 0	**18**	主机数据 15
19	地	**20**	无连接
21	DMARQ	**22**	地
23	主机 IOW	**24**	地
25	主机 IOR	**26**	地
27	IORDY	**28**	地
29	DMACK	**30**	地
31	IRQ	**32**	无连接
33	地址 1	**34**	ATA66_DETECT
35	地址 0	**36**	地址 2
37	芯片选择 0	**38**	芯片选择 1
39	活动	**40**	地
41	+5V	**42**	+5V
43	地	**44**	地

CF 卡

COMPACT FLASH 卡是一种快速存储器，体积很小，使用方便，存储量随所用的卡变化，如 256M、1G 等。CF 卡插入时只能以一个方向插入。下面是 CF 卡接口各引脚的定义：

管脚	信号名称	管脚	信号名称
1	GND	26	NC
2	D3	27	D11
3	D4	28	D12
4	D5	29	D13
5	D6	30	D14
6	D7	31	D15
7	CS0	32	CS1
8	GND	33	NC
9	GND	34	IORD
10	GND	35	IOWR
11	GND	36	+3.3V
12	GND	37	INTRQ
13	+3.3V	38	+3.3V
14	GND	39	CSEL
15	GND	40	NC
16	GND	41	RESET
17	GND	42	IORDY
18	A2	43	DMARQ
19	A1	44	DMACK
20	A0	45	NC
21	D0	46	PDIAG
22	D1	47	D8
23	D2	48	D9
24	NC	49	D10
25	NC	50	GND

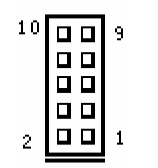
串口

COM1：**RS-232** 串行通信接口。是 **9** 芯 **D-SUB** 接口。

COM2：**RS-232** 串行通信接口。**COM2** 支持 **RS-232** 和 **RS-485**，可通过跳线 **JP2** 选择 **COM2** 的功能。

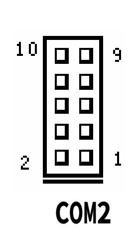
COM3/COM4/COM5/COM6：RS-232 串行通信接口，它是一个 10 芯针座接口。

COM1



COM3,COM4,COM5,COM6

管脚	信号名称
1	DCD，数据载波检测
2	RXD,接收数据
3	TXD，传输数据
4	DTR,数据终端准备完毕
5	GND，地
6	DSR，数据设置完毕
7	RTS，请求发送
8	CTS，清除发送
9	RI，响铃提示
10	未用(COM1A,COM2A,CN1)



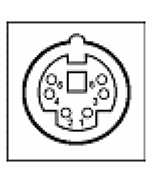
COM2

管脚	信号名称 RS232	信号名称 RS485
1	DCD，数据载波检测	RTX-
2	RXD,接收数据	RTX+
3	TXD，传输数据	
4	DTR,数据终端准备完毕	
5	GND，地	
6	DSR，数据设置完毕	
7	RTS，请求发送	
8	CTS，清除发送	X
9	RI，响铃提示	X
10	未用(COM1A,COM2A,CN1)	

键盘与鼠标接口

KM1:PS/2 键盘鼠标接口

此接口用于使用 **PS/2** 插头(**mini DIN**)的标准键盘。此接口不允许接标准 **AT** （大 **DIN**）键盘插头。你可以在标准 **AT** 键盘上使用一个 **DIN** 到 **mini DIN** 的转换器。



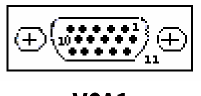
KM1

管脚	信号名称
1	键盘数据
2	鼠标数据
3	地
4	KM5V
5	键盘时钟
6	鼠标时钟

显示接口

(1) VGA 显示输出接口

这是 **15** 芯 **D** 型 **VGA** 显示器插座，可以连接所有标准 **VGA** 接口的显示器。



VGA1

管脚	信号名称	管脚	信号名称
1	红	2	绿
3	蓝	4	未用
5	地	6	地
7	地	8	地
9	VCC5V	10	地
11	未用	12	未用
13	HSYNC	14	VSYNC
15	未用		

(2) LCD 接口

这个插座用于连接 **LCD** 屏。

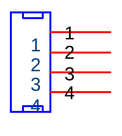
2　1

40　39

FPC1

管脚	信号名称	管脚	信号名称
1	+5V/3.3V	2	+5V/3.3V
3	GND	4	ENVEE
5	GND	6	GND
7	PD1	8	PD0
9	PD3	10	PD2
11	PD5	12	PD4
13	PD7	14	PD6
15	PD9	16	PD8
17	PD11	18	PD10
19	PD13	20	PD12
21	PD15	22	PD14
23	PD17	24	PD16
25	PD19	26	PD18
27	PD21	28	PD20
29	PD23	30	PD22
31	GND	32	GND
33	FLM	34	SHFCLK
35	LP	36	DE
37	GND	38	ENVDD
39	GND	40	VCON

电源接口



PWR1

管脚	信号名称
1	+12V
2	GND
3	GND
4	+5V

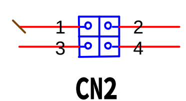
PC104 接口

ISA2 (PC104-AB1)
B1 B32
A1 A32
C1 C20
D1 D20
ISA1 (PC104-CD1)

PC104-AB1				PC104-CD1			
管脚	信号名称	管脚	信号名称	管脚	信号名称	管脚	信号名称
A1	IOCHCK	B1	GND	C1	GND	D1	GND
A2	D7	B2	RESET	C2	SBHE	D2	MEMCS16
A3	D6	B3	+5V	C3	LA23	D3	IOCS16
A4	D5	B4	IRQ9	C4	LA22	D4	IRQ10
A5	D4	B5	NC	C5	LA21	D5	IRQ11
A6	D3	B6	DRQ2	C6	LA20	D6	IRQ12
A7	D2	B7	NC	C7	LA19	D7	IRQ15
A8	D1	B8	NOWS	C8	LA18	D8	IRQ14
A9	D0	B9	+12V	C9	LA17	D9	DACK0
A10	IOCHRDY	B10	GND	C10	MEMR	D10	DRQ0
A11	AEN	B11	SMEMW	C11	MEMW	D11	DACK5
A12	A19	B12	SMEMR	C12	D8	D12	DRQ5
A13	A18	B13	IOW	C13	D9	D13	DACK6
A14	A17	B14	IOR	C14	D10	D14	DRQ6
ZA15	A16	B15	DACK3	C15	D11	D15	DACK7
A16	A15	B16	DRQ3	C16	D12	D16	DRQ7
A17	A14	B17	DACK1	C17	D13	D17	+5V
A18	A13	B18	DRQ1	C18	D14	D18	MASTER
A19	A12	B19	REFRESH	C19	D15	D19	GND
A20	A11	B20	CLK	C20	GND	D20	GND
A21	A10	B21	IRQ7				
A22	A9	B22	IRQ6				
A23	A8	B23	IRQ5				
A24	A7	B24	IRQ4				
A25	A6	B25	IRQ3				
A26	A5	B26	DACK2				
A27	A4	B27	TC				
A28	A3	B28	BALE				
A29	A2	B29	+5V				
A30	A1	B30	OSC				
A31	A0	B31	GND				
A32	GND	B32	GND				

ISA2 和 **ISA1** 都是双排管脚的插头,他们支持 **PC-104** 模块。**ISA2** 有 **64** 个管脚，**ISA1** 为 **40** 管脚。上表给出了他们的管脚定义。

辅助接口



管脚	信号名称
1	NC
2	PC104 的 RSTDRV 复位引脚
3	+12V
4	GND

GPIO 接口

GPIO 根据焊接不同的电阻可有三种定义：

* 默认定义为第(1)种

(1) 焊上 R71；不焊 R68、R69 和 R72。（此种情况为默认定义）

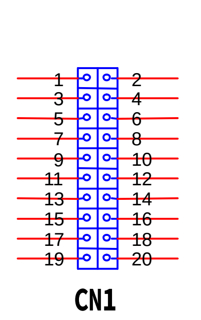
GPIO 功能为 8 输入输出（Pin3~Pin10） ＋ 7 输出（Pin11~Pin18）：

CN1

管脚	信号名称	管脚	信号名称
1	GND	2	+5V
3	IO0	4	IO1
5	IO2	6	IO3
7	IO4	8	IO5
9	IO6	10	IO7
11	OUT0	12	OUT1
13	OUT2	14	OUT3
15	OUT4	16	NC
17	OUT6	18	OUT7
19	+5V	20	GND

(2) 焊上 R68、R69；不焊 R71、R72。

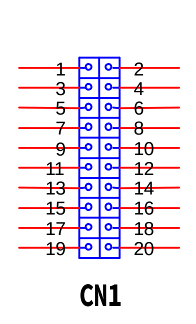
GPIO 功能为 8 输入（Pin3~Pin10） ＋ 8 输出（Pin11~Pin18）：



管脚	信号名称	管脚	信号名称
1	GND	2	+5V
3	IN0	4	IN1
5	IN2	6	IN3
7	IN4	8	IN5
9	IN6	10	IN7
11	OUT0	12	OUT1
13	OUT2	14	OUT3
15	OUT4	16	OUT5
17	OUT6	18	OUT7
19	+5V	20	GND

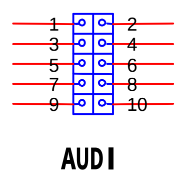
⑶ 焊上 R68、R72；不焊 R69、R71。

GPIO 功能为 8 输出（带回读）（Pin3~Pin10） ＋ 8 输出（Pin11~Pin18）：



管脚	信号名称	管脚	信号名称
1	GND	2	+5V
3	R_OUT0	4	R_OUT1
5	R_OUT2	6	R_OUT3
7	R_OUT4	8	R_OUT5
9	R_OUT6	10	R_OUT7
11	OUT0	12	OUT1
13	OUT2	14	OUT3
15	OUT4	16	OUT5
17	OUT6	18	OUT7
19	+5V	20	GND

音频接口



AUDI
01

管脚	信号名称
1	Line Out Right
2	Line Out Left
3	GND
4	GND
5	Line In Right
6	Line In Left
7	GND
8	GND
9	MIC
10	Mic Phone P

第三章

BIOS 功能简介

BIOS 简介

BIOS（**Basic Input and Output System**；基本输入输出系统）固化在 **CPU** 板上的快闪存储器中，主要功能包括：初始化系统硬件，设置各系统部件的工作状态，调整各系统部件的工作参数，诊断系统各部件的功能并报告故障，给上层软件提供软件接口控制系统硬件操作，引导操作系统等。**BIOS** 还给用户提供一个菜单式的程序接口，可用来配置各系统参数设置值，控制电源管理模式，调整系统设备的资源分配等等。

正确设置 **BIOS** 各项参数，可使系统稳定可靠地工作，同时也能获得较高的性能。不适当的甚至错误的 **BIOS** 参数设置，会使系统工作性能低下，工作不稳定，甚至无法工作。

本章内容就是关于 **BIOS** 的设置。本产品使用最先进的 **AMI BIOS** 已最大优化了系统的各项性能。

进入 BIOS 参数设定

每当接通系统电源，系统正常开机后，便可看见进入 **BIOS** 设置程序提示的信息。此时**(**其它时间无效**)**，按下提示信息所指定的按键（通常为**<Del>**键）即进入 **BIOS** 设置程序。

CMOS 中 **BIOS** 设置内容被破坏时，系统也会要求进行 **BIOS** 设置或选择所有默认设置值。

通过BIOS修改的所有设置值都保存在系统的CMOS存储器中，该 CMOS 存储器由电池供电，即使切断市电，其内容也不会丢失，除非执行清除 CMOS 内容的操作。

注意！BIOS 的设置直接影响到电脑的性能，设置错误的数值将造成电脑的损坏，甚至不能开机，请使用 **BIOS** 内定值来恢复系统正常运行。

由于本公司不断研发更新 **BIOS** 设置程序，以下的画面仅供您参考，有可能跟您目前所使用的 **BIOS** 设置程序不完全相同。

BIOS 基本功能设置

当 **SETUP** 程序启动之后，您可以看到 **CMOS Setup Utility** 主画面如下：

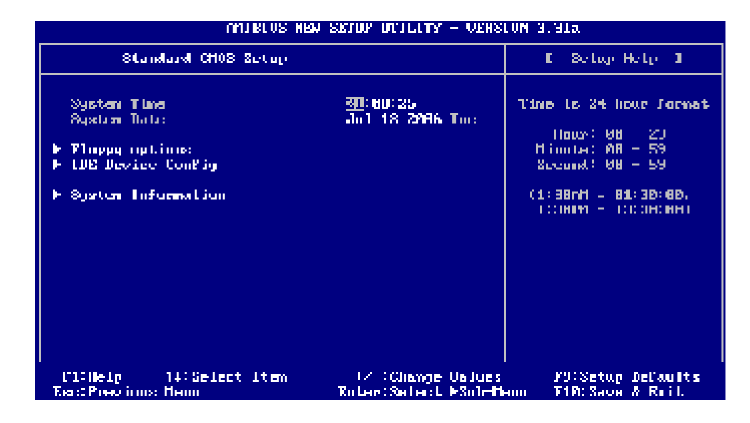


主菜单设置项的下方显示的是菜单的控制键。主菜单的底部，也就是控制键部分的下面，显示的是当前菜单中被加亮显示的选项的信息。

注意！如果您的系统在保存了您改变的设置后无法正常工作，则您可以清除 **CMOS** 内容后重新开机进入 **BIOS** 设置程序，选择 **BIOS** 中的 **Auto Configuration with Optimal Settings** 或 **Auto Configuration with Fail Safe Settings**，使用 **BIOS** 预设的各项默认值。

除非您很了解，否则不要对芯片集的默认值作任何改动。

3.1 Standard CMOS Setup



菜单底部是本菜单所用的控制键。如果你需要帮助，按**<F1>**将显示相关信息帮助您。

System Time

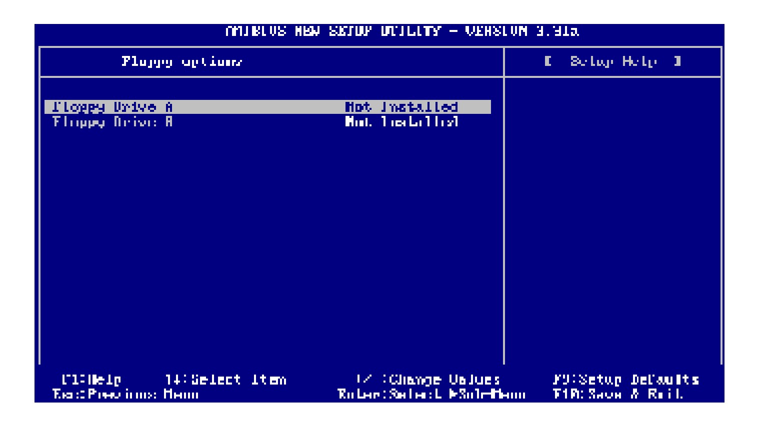
选择此选项，用**< Page Up> / < Page Down >**或是**< + > / < - >**来设置目前的时间。以时**/**分**/**秒的格式来表示。各项目合理的范围是：**Hour/**时**(00-23)，Minute/**分**(00-59)，Second/**秒**(00-59)**。

System Date

选择此项，用**< Page Up>/< Page Down >**或是**< + >/< - >**来设置目前的日期。以月**/**日**/**年的格式来表示。各项目合理的范围是：**Month/**月**(Jan-Dec), Date/**日**(01-31),Year/**年**(**最大到**2099)**。

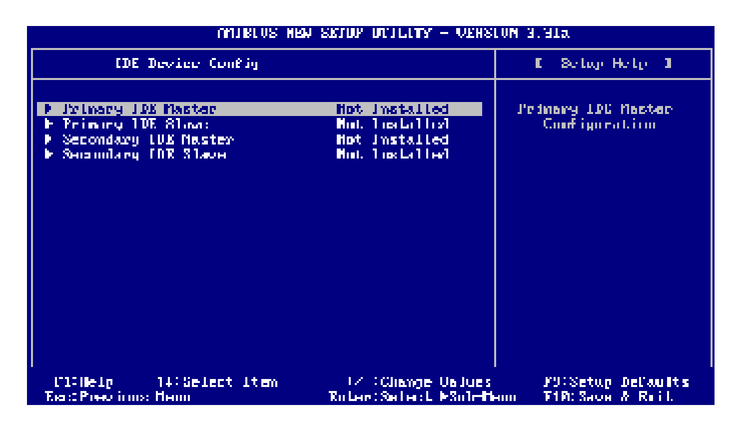
Floppy Options

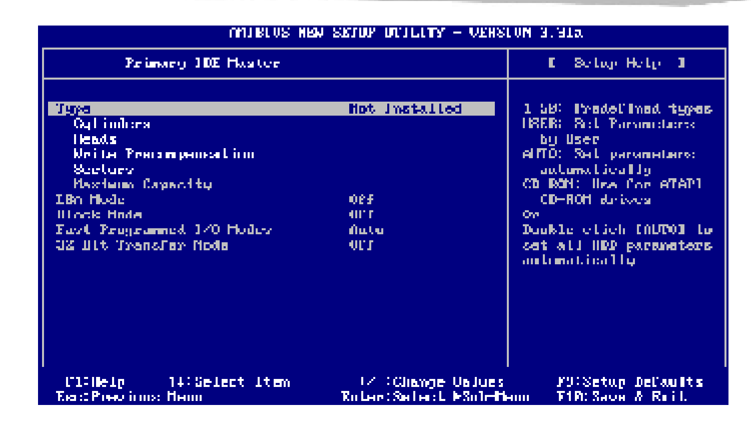
软驱接口选项，最多可接两个设备



IDE Device config

IDE设备**(**硬盘**)**配置**:**





1. Type

1～50：系统预定义的各种**IDE**参数；

USER：用户自定义**IDE**硬盘的各项参数；

AUTO：系统开机自动检测**IDE**参数；

CD/DVD：用于**ATAPI CDROM**；

ARMD：用于各种模拟**IDE**设备；

2. Cylinders

硬盘柱面参数，可设置范围：**0～65535**。

3. Heads

硬盘磁头参数，可设置范围：**0～255**。

4. Write Precompensation

用于设置硬盘写预补偿的柱面参数。

5. Sectors

硬盘扇区参数，可设置范围：**0~255**。

6. Maximum Capacity

此参数指示硬盘的最大容量。

7. LBA MODE

此选项设置为“**ON**”，可以支持大于**512MB**的硬盘，设置为“**OFF**”时适用于**NetWare**和**UINX**等操作系统。

8. BLOCK MODE

此选项设置为“**ON**”时可以提高从硬盘读写数据的速度。

9. Fast Programmed I/O MODE

在**PIO**方式下，设置此项参数可以优化硬盘时序。

10. 32 Bit Transfer MODE

此选项用于使能**32**位的硬盘访问模式，可以使硬盘读写速度达到最佳。

System Information

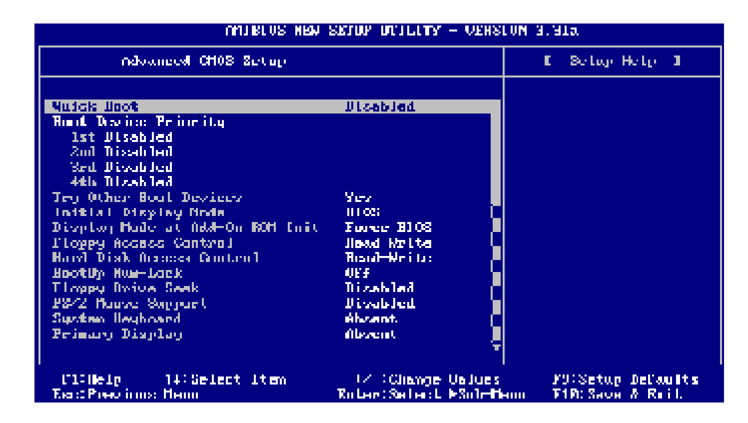
此项指示**BIOS**创建时间、处理器类型和速度，以及系统内存容量等信息。

AMIBIOS NEW SETUP UTILITY – VERSION 3.31a

System Information		[Setup Help]
BIOS Version	01/15/07	
Processor Type	Generic-X86	
Processor Speed	[illegible]	
Base Memory	[illegible]	
Extended Memory	[illegible]	
Total Memory	[illegible]	
Memory Bank 0	Standard	
Memory Bank 1	Standard	
Memory Bank 2	Standard	

F1:Help ↑↓:Select Item +/-:Change Values F9:Setup Defaults
Esc:Previous Menu Enter:Select ►Sub-Menu F10:Save & Exit

3.2 Advanced CMOS Setup




```
AMIBIOS NEW SETUP UTILITY - VERSION 3.31a
Advanced CMOS Setup                      [ Setup Help ]
PS/2 Mouse Support            Disabled
System Keyboard               Absent
Primary Display               Absent
Password Check                Setup
Boot To OS/2                  No
Wait For 'F1' If Error        Disabled
Hit 'DEL' Message Display     Disabled
System BIOS Cacheable         Disabled
C000,16k Shadow               Disabled
C400,16k Shadow               Disabled
C800,16k Shadow               Disabled
CC00,16k Shadow               Disabled
D000,16k Shadow               Disabled
D400,16k Shadow               Disabled
D800,16k Shadow               Disabled
DC00,16k Shadow               Disabled
F1:Help      ↑↓:Select Item       -/+:Change Values        F5:Setup Defaults
Esc:Previous Menu  Enter:Select ▶Sub-Menu              F10:Save & Exit
```

Quick Boot

这个功能会跳过存储器的第二、三次测试，加速**POST**的时间。而每一次的**POST**，都是一次完整的测试

Boot Device Priority

此项用于设置启动设备的优先权

Try other Boot Devices

在“**Boot Device Priority**”设置的启动设备启动失败的情况下，设置此功能选项可以让系统尝试从其它启动设备启动

Initial Display Mode

此项用于设置启动时是显示**BIOS**信息还是显示**BIOS OEM LOGO**画面。

Display Mode at Add-on ROM Init

此项用于设置**option ROM**的显示模式。

Floppy Access Control

此项用于设置软磁盘的访问权限，有**Read-Write**和**Read-Only**两个选项。

Hard Disk Access Control

此项用于设置硬盘的访问权限，有**Read-Write**和**Read-Only**两个选项。

BootUP Num-Lock

该选项用来设置开机时小键盘的状态。当设置为**ON**时，系统启动后，小键盘为数字状态；设为**OFF**时，系统启动后，小键盘为箭头状态

Floppy Drive Seek

当**Enable**时，机器启动时**BIOS**将对软驱进行寻道操作，即判断软驱为**40**磁道数（如早期的**360KB**软驱）还是**80**磁道数。

PS/2 Mouse Support

这个选项可以启用或禁止**PS/2**鼠标接口的使用。

System Keyboard

用于设置**BIOS**是否检测键盘，即使本项设为**Absent**，键盘仍能可以在**OS**中使用，只是**BIOS**不对其进行检测。

Primary Display

此项用于设置系统的显示模式，可以设置的值有：**Absent**、**VGA/EGA**、**CGA 40x25** 以及 **CGA80x25**、**Mono**。如果您用的是 **VGA** 或更高级的显示器，请选择 **VGA/ EGA**。

当设 **Absent** 时，**BIOS** 不检测 **VGA** 设备，如果没有 **VGA** 或 **VGA** 设备坏，**BIOS** 也不会报错会照常运行。

Absent	显示器不存在或无连接；
VGA/EGA	**EGA,VGA,SEGA,SVGA** 或 **PGA** 显示器适配器；
CGA 40 x25	以 **40** 行模式上电；
CGA 80 x25	以 **80** 行模式上电。

Password Check

该用于设置启动系统时要求输入普通用户密码或超级用户密码。

Boot to OS/2

设置系统引导到**OS/2**系统。

Wait For ‘F1’ If Error

该选项用来设置是否在系统启动时出现错误时显示按下“**F1**”键确认才继续进行开机，一般设置为“**Enabled**”。

Hit ‘DEL’ Message Display

该选项用来设置是否在开机时显示按下**Del**键进入**Bios**设定的提示。

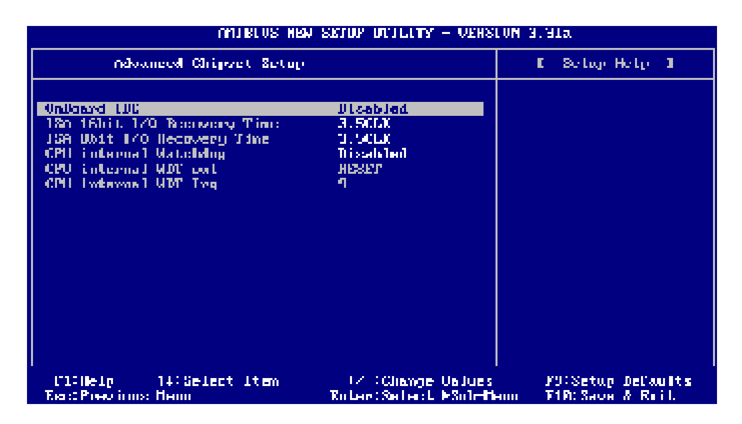
System BIOS Cacheable

该选项的缺省设置默认为**enable**，当它开启时，会提高系统**BIOS**的运行速度和改善系统的性能。

C000—DC00 Shadow

这些内存区域用来作为其他扩充卡的**ROM**映射区，一般都设定为禁止**(Disable)**。如果设**Enabled**，**BIOS**将在系统总线上搜索是否有设备响应该段地址，如果有，则将设备内容拷贝到对应的内存，之后**CPU**直接访问该段地址的内存，而不访问总线上的设备。

3.3 Advanced Chipset Setup



OnBoard IDE

该项设置用来控制主板上**IDE**接口是否可用，有**Both**，**Disabled**，**Primary**和**Secondary**四种选择。

ISA 16bit I/O Recovery Time

该项用于设置**ISA**总线**16 bit I/O**访问时的恢复时间。

ISA 8bit I/O Recovery Time

该项用于设置**ISA**总线**8 bit I/O**访问时的恢复时间。

CPU internal Watchdog

该项设置**CPU**内部的看门狗的使能和禁止。

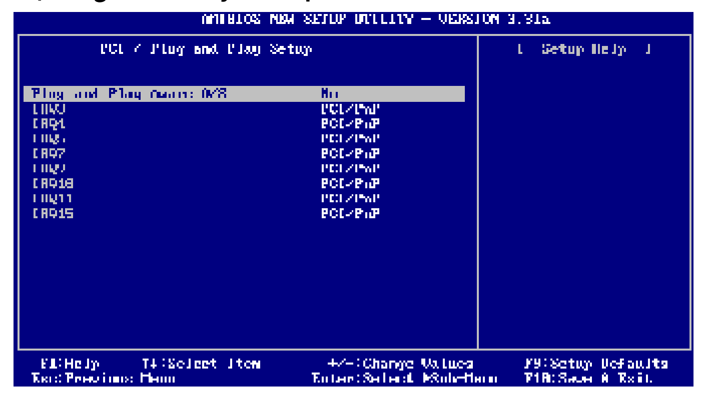
CPU internal WDT out

CPU内部看门狗的输出所产生的动作。有**Reset**和**Irq**两个选项，用于设置看门狗定时器溢出时复位**CPU**或者产生中断。

CPU internal WDT Irq

设置**CPU**内部看门狗所使用的中断号。

3.4 PCI/Plug and Play Setup



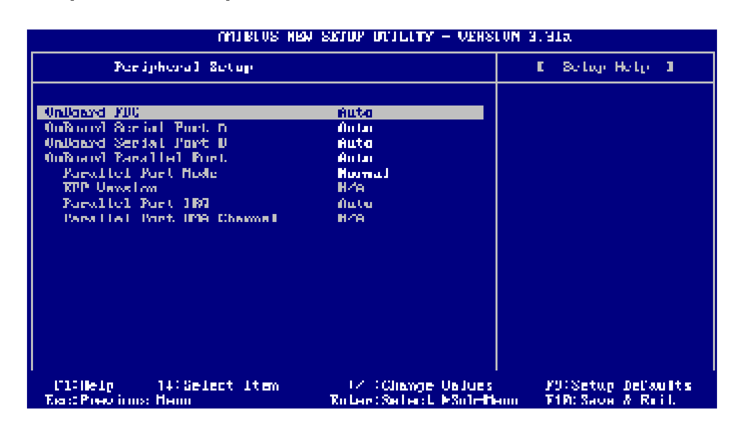
Plug and Play Aware O/S

该项用来选择所装操作系统是否为**PNP**（即插即用）操作系统。当设为**NO**时，**BIOS**将负责所有的**PNP**工作。建议选用该值。

IRQ 3～15

本项目用以指定**IRQ**中断是**PNP**方式还是保留给**ISA**使用。

3.5 Peripheral Setup



OnBoard FDC

如果你的系统主板上安有软驱控制器（**FDC**），你又希望使用它，选择 **Enabled** 或 **Auto**。

Onboard Serial PortA

该项设置用来配置 **CPU** 板上第一个串行接口的类型，并对中断和 **I/O** 地址作响应分配。有 **Auto**（**BIOS** 自动配置）、**Disabled(**禁止不用**)**，**3F8/COM1**、**2F8/COM2**、**3E8/COM3**、**2E8/COM4**。

Onboard Serial PortB

该项设置用来配置 **CPU** 板上第二个串行接口的类型，并对中断和 **I/O** 地址作响应分配。有 **Auto**（**BIOS** 自动配置）、**Disabled(**禁止不用**)**，**3F8/COM1**、**2F8/COM2**、**3E8/COM3**、**2E8/COM4**。

OnBoard Parallel Port

该项用来配置并行口所用的的中断及 **I/O** 口地址。**AUTO** 为 **BIOS** 自动控制。

Parallel Poart Mode

该项设置指定并行口的工作模式：**Normal**、**Bi-Dir**、**EPP**、**ECP**。**Normal** 表示单向数据传输的正常速度；**Bi-Dir** 表示双向数据传输的正常速度；**EPP** 表示双向数据传输下的最大速度；而 **ECP** 表示在双向数据传输下比 **EPP** 更快的速度。

EPP Version

该选项用于选择增强型并行接口 **EPP** 的版本。

Parallel Poart IRQ

该选项用于选择并行接口所用的中断。

Parallel Poart DMA Channel

该选项用于选择并行接口所使用的 **DMA** 通道。

3.6 Change User&Supervisor Password

当按**Change User/ Supervisor Password**后，在对话框中输入新的密码后，此栏会显示用户密码已安装。



3.7 Auto Configuration with Optimal Settings

此菜单用于在你的系统配置中装入缺省值。这些缺省设置是最优的，可以发挥所有硬件的高性能。



3.8 Auto Configuration with Fail Safe Settings



该选项的功能是将各项设置初始化为实现最基本的和最安全的系统功能的值。要执行此项功能，先选中此选项按下**< Enter >**即可，接着系统会在屏幕上显示出要您确认的信息，按下**< Enter >**键确定执行该项功能。

3.9 Save Settings and Exit



当您完成了所有的修改操作，想将原来的设置参数覆盖掉时，可执行此项功能，新的设置参数将保存在**CMOS**的存储器中。要执行此操作，先选定此选项并按下**< Enter >**键，再按**< Enter >**键即可。

3.10 Exit Without Saving



当您所做的任何更改设置的动作不想存入 **CMOS** 的存储器中，可先选定此选项并按下**< Enter >**键，再按**<Enter >**键即可。

附录

驱动程序的安装

USB 驱动程序的安装

以下为在 **DOS** 环境下的安装步骤：

(1) 启动电脑，进入 **DOS** 环境 **à** 从驱动盘中找到三个文件，它们是：**USBASPI.SYS**、**USBCD.SYS**、**NJ32DISK.SYS**，将这三个文件复制到 **C** 盘根目录（即 **C:**）下；

(2) 修改 **CONFIG.SYS** 文件，在该文件中增加如下内容：

```
DEVICE=HIMEM.SYS
DOS=HIGH,UMB
LASTDRIVE=Z
DEVICEHIGH=USBASPI.SYS /v /o
DEVICEHIGH =NJ32DISK.SYS
DEVICEHIGH =USBCD.SYS /d:USBCD001
```

(3) 修改 **AUTOEXEC.BAT** 文件，在该文件中增加如下内容：

```
LH MSCDEX /d:USBCD001 /l:d
```

(4) 当修改完成后，需重新启动才能使新装的驱动程序生效。

注:(1) 必须先将 **USB** 设备插入 **USB** 接口，然后再启动电脑，系统才能识别 **USB** 设备。

(2)USB 设备的盘符在硬盘盘符之后。

(3) 其中：**DEVICEHIGH =USBCD.SYS /d:USBCD001**

LH MSCDEX /d:USBCD001 /l:d

这两行是对 **USB** 光驱的支持。如果不使用 **USB** 光驱，这两行代码可省略。

网络驱动程序的安装

驱动盘中提供了 **DOS** 环境下的网络驱动程序 **R6040PD.COM**,即 **DOS** 下网卡的 **Packet Driver**。关于该程序的用法也可参见它的联机帮助。

(1) **R6040PD.COM** 的使用

① **R6040PD.COM** 的安装

在 **DOS** 提示符下键入以下命令：

>R6040PD [options] <packet_int_no>

例子：

R6040PD –slot=1 -m=00:90:00:2a:00:01 0x66

-slot=1 ：选择网络接口 **LAN1**

-m ：设定 **MAC** 地址为 **00:90:00:2a:00:01**；

0x66 ：使用中断 **0x66**；

其中**<packet_int_no>**为指定给 **Packet Driver** 的中断号，该中断号即作为 **Packet Driver** 的入口。可以选用的中断号的范围为：**0x60 à 0x66**，**0x68 à 0x6f** 和 **0x78 à 0x7e**。

注：**-slot=1** ：安装 **LAN1** 接口的驱动程序；

② **R6040PD.COM** 的卸载

卸载此 **packet driver** 时使用参数**-u**，并且需要指定 **slot** 号和中断号，示例如下：

★ 安装 **LAN1** 的网络驱动

>r6040pd –slot=1 –m=00:90:00:2a:00:01 0x60

★ 卸载 **LAN1** 的网络驱动

>r6040pd –slot=1 -u 0x60

(2) 在 **IPX** 协议（**Novell**）下 **Packet Driver** 的安装步骤：

① 在 **DOS** 提示符下装入 **LSL.COM**

>LSL

② 在 **DOS** 提示符下装入 **Packet Driver**

>R6040PD <packet_int_no>

③ 在 **DOS** 提示符下执行 **PDIPX.COM**

>PDIPX

④ 调用 **VLM**

其中 **PDIPX.COM** 是 **Intel** 开发的一个 **IPX** 驱动程序，它可以和网卡的 **Packet Driver** 通信，从而可以在 **Packet Driver** 上使用 **IPX** 协议。

(3) 在 **TCP/IP** 协议下 **Packet Driver** 的安装步骤：

A、在 **DOS** 下的安装步骤：

① 在 **DOS** 提示符下装入 **Packet Driver**

>R6040PD <packet_int_no>

② 现在可以运行使用 **TCP/IP** 协议的 **DOS** 应用程序了。

B、在 **Windows 3.x** 增强模式下的安装步骤：

① 在 **DOS** 提示符下装入 **Packet Driver**

>R6040PD <packet_int_no>

② 在 **DOS** 提示符下执行 **WINPKT.COM**

>WINPKT <packet_int_no>

这里的中断号**<packet_int_no>**必须和第①步里所指定的中断号**<packet_int_no>**相同。

③ 进入 **Windows**，并运行 **Winsock** 程序，**Winsock** 使用的中断号 **<packet_int_no>** 也必须和第①步里所指定的中断号 **<packet_int_no>**相同。

④ 现在可以运行使用 **TCP/IP** 协议的 **Windows** 应用程序了。

其中 **WINPKT.COM** 提供了 **Windows 3.x** 增强模式下的应用程序和网卡的 **Packet Driver** 之间的一个接口。从而可以在 **Packet Drive** 上运行相关的 **Windows** 应用程序。

数字 I/O 驱动程序的安装

以下用 C 语言形式描述了数字 IO 设备的编程，它分为如下步骤：

（1） 映射基址到端口。

（2） 读写端口。

（3） 取消映射。

```
unsigned long _pascal in_32(unsigned int io_port)
{
        unsigned long ReturnValue;
        _asm{
                    mov dx,io_port
                    in    eax,dx
                    mov ReturnValue,eax
                }
        return ReturnValue;

}
void _pascal out_32(unsigned int io_port, unsigned long io_data)
```

```
{
    _asm{
            mov dx, io_port
            mov eax, io_data
            out dx,    eax
        }
}

int CS1_IO_map(unsigned short port_base_address)
{
    unsigned long address,value;

    // set CS1 mode.
    out_32(0xcf8,0x80003887);
    value = in_32(0xcfc);
    value &= 0xfdffffff;            // bit25=0
    out_32(0xcfc,value);

    // map base address to IO port.
    address = (unsigned long)port_base_address;
    address |= 0x00000001;          // bit0=1
    address &= 0xfffffff9;          // bit2=0,bit1=0
    out_32(0xcf8,0x80003890);
    out_32(0xcfc,address);

    return 0;
}

int CS1_unmap(void)
{
    unsigned long value;
```

```
	out_32(0xcf8,0x80003890);
	value = in_32(0xcfc);
	value &= 0xfffffffe;//bit0 = 0
	out_32(0xcfc,value);

	return 0;
}

int CS1_GPIO_control(unsigned long base_address,unsigned char data)
{
	unsigned short port_address;

	port_address = base_address + 0x0000;
	outportb(port_address,data);

	return 0;
}

int CS1_GPIO_in(unsigned short base_address, unsigned char *pdata)
{
	unsigned short port_address;

	if(pdata == NULL)
	{
		printf("invalid parameters!\n");
		return -1;
	}

	port_address = base_address + 0x0001;
	*pdata = inportb(port_address);
```

```
        return 0;
}

int CS1_GPIO_out(unsigned short base_address,unsigned char data)
{
        unsigned short port_address;

        port_address = base_address + 0x0001;
        outportb(port_address,data);

        return 0;
}
```

用户操作数字 **IO** 时，先用 CS1_IO_map（）进行端口映射（可用系统中任意空闲的端口），再用 CS1_GPIO_control（）、CS1_GPIO_in（）、CS1_GPIO_out（）执行需要的操作，操作完后用 CS1_unmap 取消映射。

注意：**IO0 ~ IO7** 作输入使用时，应该在没有设置输入信号时先设置 **OUT5** 指明是读还是写（**1** 为写，**0** 为读），然后再设置输入信 号（即高、低电平）进行读操作，否则可能损坏硬件。

Watchdog 编程指引

(1) Super I/O 看门狗

Super I/O芯片提供一个可按分或按秒计时的，最长达**255**级的可编程看门狗定时器**(**以下简称**SIOWDT)**。通过编程，**SIOWDT**超时事件可用来将系统复位或者产生一个可屏蔽中断。

以下用**C**语言形式描述了**SIOWDT**的编程。必须注意：在对**SIOWDT**

进行操作之前，需先进入**SIOWDT**编程模式；在结束对**SIOWDT**的操作之后，退出**SIOWDT**。

对**SIOWDT**的编程需遵循以下步骤：

» 进入**SIOWDT**编程模式

» 设置**SIOWDT**工作方式/启动**SIOWDT**/关闭**SIOWDT**

» 退出**SIOWDT**编程模式

SIOWDT的编程方法，请参看以下示范代码：

```
#define INDEXP  0x2e
#define DATAP       0x2f
//Super I/O Watchdog
#define STARTPROG {outportb(INDEXP,0x87);
                outportb(INDEXP,0x87);}
#define ENDPROG outportb(INDEXP,0xaa);
#define SELEDEV(x) {outportb(INDEXP,7);
outportb(DATAP,x);}
#define WRITEREG(reg,val) {outportb(INDEXP,reg);
outportb(DATAP,val);}

//1.Initial Watchdog device
short SIOWTD_Setup(short irq)
/*irq=3,4,5,6,7,9,10,11,12,13,15,0:disable interrupt, 0xff:reset*/
{
unsigned char oldval;
//start programming Watchdog
STARTPROG
```

```
//Active Watchdog Device
SELEDEV(8) //logical device 8
WRITEREG(0x30,0x01)

outportb(INDEXP,0x2b);
unsigned char oldval=inportb(DATAP);

if(irq==0xff) //WatchDog cause System Reset
{
oldval &= 0xef; //BIT4=0
WRITEREG(0x2b, oldval);
}
else //Watchdog cause System Interrupt
{
oldval |= 0x10; //BIT4=1
WRITEREG(0x2b,oldval)
WRITEREG(0xf7,irq)
}
//end programming watchdog
ENDPROG
return 0;
}
//2.start Watchdog to count
short SIOWTD_Enable(short time,short unit)
/*unit=0:second, =1:minutes */
{
```

```
if(time<1 || time>255) return -1;
if(unit<0 || unit>1) return -1;
//start programming watchdog
STARTPROG
SELEDEV(8) //logical device 8
//select Watchdog Timer clock
switch(unit)
{
case 0:
WRITEREG(0xf5,0x00) //BIT3=0,secondes
break;
case 1:
WRITEREG(0xf5,0x08) //BIT3=1,minutes
break;
}
WRITEREG(0xF6,time) //set timeout value
//end programming watchdog
ENDPROG
return 0;
}
```

(2) CPU 看门狗

CPU芯片内部提供了一个8级的可编程看门狗定时器(以下简称CPUWDT)。通过编程，CPUWDT超时事件可用来将系统复位或者产生一个可屏蔽中断。8级分别为81.92us、83.9ms、167.8ms、335.5ms、671.1ms、1.34s、2.68s和5.37s。

CPUWDT的功能在**BIOS**中设置，详细情况参见本手册第三章中的**3.3 Advanced Chipset Setup**一节的内容。

以下用**x86**汇编语言形式描述了**CPUWDT**的编程。**CPUWDT**的编程方法，请参看以下示范代码：

```
//选定CPU看门狗控制寄存器
mov eax, 80003844h
mov dx, 0cf8h
out dx, eax
//设定看门狗超时时间，并启动看门狗以开始计时
mov dx,0cfch
in  eax,dx
or  eax,00800000h       ；bit23=1:启动看门狗
                        ；如果bit23=0,则禁止看门狗
and eax,ffffff00h
or  eax,00000001h       ；bit[7:0]=01h,定时时间为81.92us
                    ；如果bit[7:0]=02h,定时时间为83.9ms
                    ；如果bit[7:0]=04h,定时时间为167.8ms
                    ；如果bit[7:0]=08h,定时时间为335.5ms
                    ；如果bit[7:0]=10h,定时时间为671.1ms
                    ；如果bit[7:0]=20h,定时时间为1.34s
                    ；如果bit[7:0]=40h,定时时间为2.68s
                    ；如果bit[7:0]=80h,定时时间为5.37s
out dx,eax          ；将设定值写入寄存器，看门狗开始工作
```

PC104 总线设备可用的 I/O 口和内存空间

系统板已经占用了部分 **I/O** 口地址，**PC104** 扩展总线可用的 **I/O** 口地址范围为：**0x200-0x27F**，**0x300-0x340**，**0x280-0x2C0**。

PC104 扩展总线可用的内存空间如下：

C8000 – **CFFFF : 32KB;**

D0000 – **DFFFF : 64KB;**

E0000 – **E7FFF : 32KB;**

注：部分软件会占用以上内存段，如 **EMM386.exe** 等。

PC104 扩展总线不支持 **DMA** 访问模式。

BIOS 在线刷新工具软件的使用方法

驱动盘中提供了 **DOS** 环境下的在线刷新 **BIOS** 的工具软件 **flash.exe**。

Flash.exe 的使用方法如下：

⑴ 保存原 **BIOS**。

在 **DOS** 提示符下键入：

>flash.exe /Sxxxx.xx

即可将主板上的 **BIOS** 备份到文件 **xxxx.xx** 中，其中 **xxxx.xx** 为文件名。

注：**/S** 与 **xxxx.xx** 之间没有空格。

⑵ 刷新 **BIOS**。

在 **DOS** 提示符下键入：

>flash.exe xxxx.rom

即可将新 **BIOS** 文件 **xxxx.rom** 写入主板 **BIOS** 中，其中 **xxxx.xx**

为新 **BIOS** 文件的文件名。

I/O 口地址映像表

系统的每一个外设都被分配了一套 **I/O** 口地址，也成为了设备的身份识别。总共有 **1K** 可用地址空间。下表给出了用于工业级主板的 **I/O** 口地址。

地址	设备描述
000h - 00Fh	**DMA** 控制器**#2**
020h - 021h	可编程中断控制器**#1**
040h - 043h	系统计时器
061h	**NMI**
060h,062h-064h	标准 **101/102** 键盘控制器
070h - 071h	实时时钟**,CMOS Memory**
080h - 09Fh	**DMA** 页寄存器
092h	系统控制寄存器
0A0h - 0A1h	可编程中断控制器**#2**
0C0h - 0DFh	**DMA** 控制器**#1**
170h - 177h	从 **IDE**
1F0h - 1F7h	主 **IDE**
2F8h - 2FFh	串行端口 **#2(COM2)**
378h - 37Fh	并行端口 **#1(LPT1)**
3B0h - 3DFh	显示卡接口
3F0h - 3F5h	标准软磁盘控制器
3F8h - 3FFh	串行端口 **#1(COM1)**
0CF8h	**PCI** 组态地址寄存器端口
0CFCh	**PCI** 组态数据读写端口

中断请求线（IRQ）

主板上总共有 **15** 条可用的 **IRQ** 线。外设通过中断请求线向 **CPU** 申请所需的服务。下表给出了工业级主板上设备所用的 **IRQ**。

级别	功能
IRQ 0	系统定时器
IRQ 1	标准 **101/102** 键或 **Microsoft** 键盘
IRQ 2	可编程的中断控制器
IRQ 3	串口**#2**
IRQ 4	串口**#1**
IRQ 5	保留
IRQ 6	软盘控制器
IRQ 7	并口**#1**
IRQ 8	系统 **CMOS/**实时钟
IRQ 9	保留
IRQ 10	保留
IRQ 11	保留
IRQ 12	保留
IRQ 13	保留
IRQ 14	保留
IRQ 15	保留

另外，主板上还集成了两个 **USB** 和一个 **LAN** 等 **PCI** 设备。**PCI** 设备的中断是由系统根据情况动态分配的，系统每次上电开机都列出以上设备实际分配的中断号。

部分中断可通过设置 **BIOS** 强制保留为 **ISA/PC104** 总线设备使用，而不参与 **PCI** 设备的分配过程，详细设置方法可参考 **BIOS** 设置 **PCI/PnP** 部分。